

A Mutual Information Based Approach to Optimising View Orientation for Direct Volume Rendering

Richard Cartwright, Minsi Chen, Richard Hill

Department of Computer Science, University of Huddersfield, Huddersfield, UK

Abstract—When simulating and visualising volumetric data, we often do not have the means to control the visualisation. This can be due to performance-cost or usage with an HPC where no graphical output is available. We propose a method to measure and determine a metric for the current visualisation's view, using the mutual information shared by the volumes raw simulated data and the visualisation's view achieved directly in the CUDA kernel during simulation.

Only the view's rotation around a sphere is considered whilst maintaining all other degrees of freedom at a constant. A promising result is achieved showing clear/notable areas around the visualisation. We also discuss further work to improve the effectiveness of the visualisation metric.

Index Terms—Mutual information, Volume Rendering, Visualisation

1 INTRODUCTION

DATA visualisation plays an invaluable role in presenting and interpreting data from scientific simulations. An ability to, without manual intervention, steer a simulation visualisation to display an optimal view at certain simulation steps can serve as valuable feedback. In order to achieve a level of automation in generating optimal views of a simulation, we need quantitative metrics that supports the measurement of view optimality. One of these metrics – Shannon's information theory [1] – has been widely applied to telecommunication and information compression. Its application to data visualisation has motivated exploration into the quantification of information embedded in visual representations of data [2]. We examine the specific scenario whereby in-situ data, which varies in real-time, requires visualisation to assist the comprehension of complex phenomena that might lie within.

We present our study on the use of mutual information (MI) for quantifying the similarity between a time-varying volumetric dataset and its visualisation. We use the term *similarity* to denote how closely a visual representation depicts the important characteristics of its underlying data.

2 RELATED WORK

Simulation data properties are often unpredictable and varied. Visualised data, or the context within which the data is produced, can change and develop over time. Noteworthy advancements could be occurring within the data which are not necessarily the main focus in the screen space; this can result in data being undetected by the receiver. Shannon's original research [1], stated that information loss can be regarded as *noise* or simply a risk to successful transmission. Applying this to data visualisation, we can regard visualisation as a means of data transmission. In this context, information loss can be quantified against viewport configuration and the dynamic changes in data.

The idea of using visualisation view-space to assist the process of visualisation is commonly referred to as *viewpoint information*. Essentially, there are two choices: change something within the visualisation, relative to the viewpoint [3], [4], [5]; or change the viewpoint itself to improve the visualisation quality [6], [7], [8]. Whilst this article focuses primarily on the latter, both approaches share a similar trait in that they predominantly use information entropy-based measurement to quantify the optimal result, using the viewpoint to determine the optimal view.

Castelló et al. [9] used *viewpoint mutual information* (VMI) as a tool for intelligently decimating a mesh, allowing the VMI to work as a metric for measuring the error. This method produced an improved visual result than using general geometric simplification calculations by removing interior vertices, which reduces triangle count and mesh complexity. Although aimed more towards the video game optimisation field, the method of using VMI to simplify a dataset is beneficial to general data visualisation.

Bordoloi and Shen [10] proposed a measurement for *viewpoint "goodness"* regarding the viewpoint's position in screen space, giving users the ability to define an importance function so that the visualisation can aim to show as many important voxels as possible. Their technique was shown to be effective for large static datasets.

Ruiz et al. [11] determined the mutual information of the view and the voxel as a *viewpoint information channel* and use it as a measure of viewpoint goodness. The VMI was computed based on individual projected voxels from a given viewpoint. In addition to using voxel visibility values for view manipulation to determine informative viewpoints, ambient occlusion was used to shade voxels identified as important to the visualisation. One of the noticeable disadvantages of their method was the multiple rendering passes required for calculating viewpoint information.

Our proposed method differs to Ruiz et al. by evaluating MI on a per-ray basis instead of per voxel. This resulted a volume rendering algorithm that only requires one pass for the computation of viewpoint information (see Algorithm 1). We consider this to be

crucial improvement when visualising time-varying data.

3 METHODS

3.1 Definition of Mutual Information

The entropy of a random variable X measures the amount of information carried by the variable. Evaluating entropy requires the probability density function of a given random variable; see Equation 1.

$$H(X) = - \sum_{x \in X} P(x) \log P(x) \quad (1)$$

Similarly, the information carried by two random variables X and Y , that coincide is dependent on its joint probability as formulated in Equation 2.

$$H(X, Y) = - \sum_{x \in X, y \in Y} P(x, y) \log P(x, y) \quad (2)$$

Using both the respective and joint entropy values, mutual information gives the ability to estimate the amount of information that two variables share [12] as shown in Equation 3.

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (3)$$

In the context of data visualisation, X denotes the raw data resulting from a simulation and Y denotes visual elements representing the data (e.g. pixels in the final image). As mutual information measures the degree of dependency between two random variables, we can exploit this metric to quantify how closely a visualisation represents its dataset.

3.2 View Space Mutual Information

The orientation of view space is defined by the relative camera pose to a volume dataset, which is denoted in spherical coordinates, i.e. θ for yaw, ϕ for pitch and r for the distance. Furthermore, the

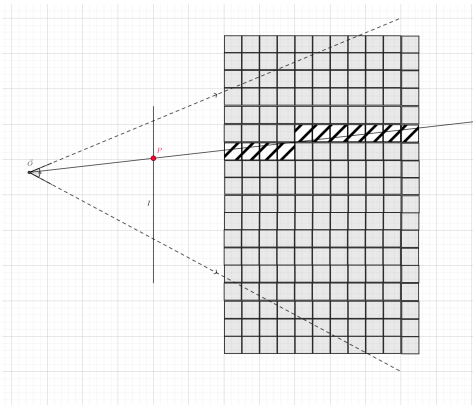


Fig. 1: Spatial relationship between the view space and the volume data.

rendered image resolution and focal length determine field of view and ray direction.

To quantify the information contained in a volume rendered dataset, we evaluated the mutual information in the view space.

Let $X = \{\vec{x}_1, \dots, \vec{x}_n\}$ denote a set of voxels, and $P = \{\vec{p}_1, \dots, \vec{p}_n\}$ is a set of pixels on the image plane. Y is the set of voxels “visited” by the ray casting process. We regarded the visualised data Y as a subset of X that are seen by rays cast through the image plane P , see Figure 1.

Thus, the mutual information of a dataset and its visualisation is:

$$I(X; P) = H(X) + H(P) - H(X, P) \quad (4)$$

The evaluation of $H(X, P)$ in Equation 4 requires the joint probability $P(X \cap P)$, where $X \cap P$ is a subset of X that is within the viewing space:

$$Y = X \cap P = \{\vec{x} \in X : \exists \vec{p} \in P \wedge d(\theta, \phi, \vec{x}, \vec{p}, \vec{o}) = 0\} \quad (5)$$

The function d is the ray intersection test.

$$d(\theta, \phi, \vec{x}, \vec{p}, \vec{o}) = 1 - [(\mathbf{R}(\theta, \phi)\vec{x} - \vec{o}) \cdot (\vec{p} - \vec{o})] \quad (6)$$

\mathbf{R} is the rotation matrix transforming \vec{x} into the view space.

The optimisation of viewing parameters can be modelled as a maximisation problem on the mutual information.

$$(\theta, \phi) = \arg \max_{(\theta, \phi) \in [-\pi, \pi]} I[X; P(\theta, \phi)] \quad (7)$$

Equation 7 is a specific MI cost function defined over the view orientation variables.

3.3 Implementation and Experiments

The probability density in Equation 7 was approximated using histograms. In this paper, we used histograms containing 512 bins which were directly populated in the CUDA kernel during simulation during the volume rendering process, see Algorithm 1. The methodology to bin each collision point during the ray march is an important distinction when compared to similar methods, limiting the application to volume rendering applications. The rationale behind this approach is to compute the probability density for the entire view of the volume rather than solely on the pixel data within the frame-buffer.

4 RESULTS AND DISCUSSION

This section presents preliminary evidence that the mutual information of the output of a volume renderer and raw simulation data can be used to determine the optimal way of viewing a simulation.

All results and data presented here were collected using the output of a modified version of HemeLB [13]. We added a CUDA-based ray-tracing kernel to the original HemeLB in order to fully parallelise the simulation and visualisation processes.

4.1 Mutual Information Over Time

A full blood-flow simulation was performed and recorded using only the default view of the visualisation, $\theta = 0; \phi = 0$. In Figure 2a, the simulation plateaus at the 10,000th timestep before falling at the 40,000th step, where the simulation begins to stabilise.

Figure 2c shows a *busy* looking simulation paired with a high (plateaued) amount of mutual information when paired with the simulation data, whereas Figure 2d displays a more *stable* visualisation paired with a lower (further declining) mutual information. This result tells us that Figure 2c is sharing more information with the simulation data whilst Figure 2d is sharing less. Due to the previously discussed limitations of data visualisation, it is likely that there will be some kind of information loss when visualised, so disregarding the time-step displayed in Figure 2d could potentially be unwise. The comparison of varying visualisations paired with the same time-step should prove a useful resource.

Algorithm 1 Pseudo code for the CUDA single-pass ray casting and histogram building kernel.

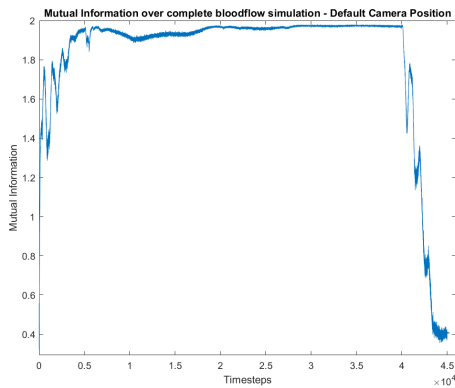
Require: *VolumeData, TransferFunction, ImageWidth, ImageHeight, ColourHistogram*

Ensure: *OutputColour, ValueHistogram*

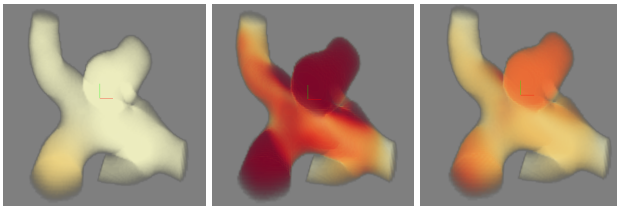
```

1: procedure VOLUMERENDERKERNEL( $x, y, S, \delta s, \rho, \mathbf{M}$ )  $\triangleright x, y$  are pixel position;  $S$  is ray marching step;  $\delta s$  is ray step size;  $\rho$  is voxel
   density;  $\mathbf{M}$  is the view matrix
2:    $OutputColour \leftarrow 0$ 
3:    $Ray \leftarrow (\mathbf{o}, \mathbf{d}')$   $\triangleright$  Ray defined in view space
4:    $t \leftarrow \text{INTERSECTVOLUME}(VolumeData, Ray)$ 
5:   if  $0 \leq t < MAX$  then  $\triangleright$  Ray interests with the nearest voxel in Volume
6:      $\mathbf{p} \leftarrow \mathbf{o} + \mathbf{d}'t$   $\triangleright$  Position of the nearest voxel
7:      $\delta \mathbf{d} \leftarrow \mathbf{d}'\delta s$   $\triangleright$  Ray cast step vector
8:      $s \leftarrow 0$ 
9:     while  $s < S$  do  $\triangleright$  March along the ray for  $S$  steps
10:       $Value \leftarrow \text{SAMPLEVOLUME}(p)$   $\triangleright$  Sample the value at  $\mathbf{p}$ 
11:       $ValueHistogram \leftarrow \text{BINVALUE}(Value, ValueHistogram)$   $\triangleright$  Add the simulation result to histogram bin
12:       $Colour \leftarrow \text{TRANSFERFUNCTION}(Value)$   $\triangleright$  Transfer  $Value$  to RGBA
13:       $Colour \leftarrow Colour \cdot \rho$   $\triangleright$  Translucency is determined by  $\rho$ 
14:       $ColourHistogram \leftarrow \text{BINVALUE}(Colour, ColourHistogram)$   $\triangleright$  Add ray's colour result to the correct histogram bin
15:       $OutputColour \leftarrow \text{ALPHABLEND}(OutputColour, Colour)$   $\triangleright$  Integrate voxel colours
16:       $\mathbf{p} \leftarrow \mathbf{p} + \delta \mathbf{d}$   $\triangleright$  March along the ray by  $\delta \mathbf{d}$ 
17:       $s \leftarrow s + 1$ 

```



(a) Graph displaying mutual information over full simulation



(b) Step 100 (c) Step 20,000 (d) Step 41,000

Fig. 2: Graph displaying the mutual information between the simulated data and the visualisation output from the volume renderer over a full blood flow simulation (45,246 steps total)

4.2 Mutual Information Over Views

The mutual information shared between visualisation and simulation data shows little relevance to different time-steps during the simulation. However, mutual information shared during the *same* time-step could indicate the quality of the visualisation. We demonstrate the results of the mutual information metric of a single time-step, specifically step 20,000. Figure 3a displays a surface with the axis representing the azimuth and elevation relating to a

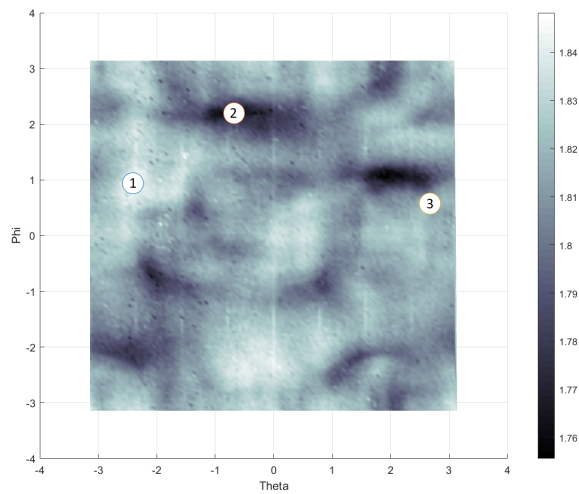
sphere around the geometry with a colour map showing the mutual information of that point.

Figure 3a displays the mutual information on a sphere around the geometry, with white representing high MI and black representing low MI. Predictably, the mutual information values change around the geometry in a controlled fashion, showing distinguishable areas of both low and high values. Due to the busy nature of the visualisation and simulation stage, the mutual information around the geometry leans towards the higher end of the scale shown by the large amounts of lighter areas on the surface (> 1.82). This behaviour displays a well-defined grouping of mutual information values that can potentially aid in optimisation of the algorithm.

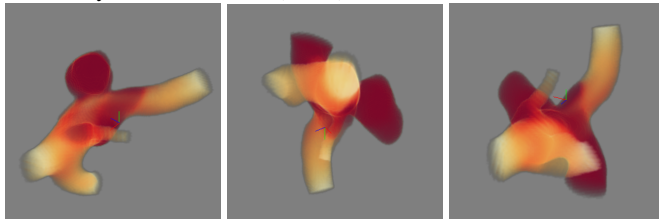
The mutual information variation between the minimum and maximum is ~ 0.08 bits of information shared between the visualisation and simulation data. Whilst this difference is relatively small, notable differences can be seen in the figures displayed in Figure 3. The minimum mutual information visualisation (Figure 3c) shows a visual representation with much of the shown geometry being low on the transfer function. When compared with the maximum mutual information view (Figure 3b), it is clear that small parts of the geometry are not visible within the volume as well as a portion of medium-pressure areas partially blocked by the large area of low-pressure. The view containing the most information is Figure 3b where it is clear that much of the volume's geometry is visible, predictably displaying an a seemingly effective overview of the volume due to it having the closest amount of information shared with the simulation data.

Figure 3d displays a randomly selected midmost ($MI = 1.810$) mutual information. This view appears to show a similar *focused* view to the low mutual information view (Figure 3c); however less low-pressure areas are present and the small vessel (mentioned above) is present. Whilst difficult to directly quantify, it is apparent that Figure 3b gives a pleasing overall view of the data, displaying the full width of the volume, as well as much of the depth when directly compared with Figure 3c and Figure 3d.

Currently, the naive method used to gather the mutual information of views is slow, taking ~ 20 minutes to gather 36^2 points



(a) Surface displaying mutual information over a sphere containing the volume, mapped in radians $[-\pi, \pi]$. Marker 1 indicating the peak mutual information (1.845), marker 2 the lowest (1.763) and marker 3 a randomly selected median (1.810).



(b) View containing highest mutual information for step 20,000 (Marker 1) (c) View containing lowest mutual information for step 20,000 (Marker 2) (d) View containing the median mutual information for step 20,000 (Marker 3)

Fig. 3: Figures displaying three samples of visualisations during step 20,000. The minimum, maximum and middle mutual information visualisations are presented

around a sphere, this is clearly far from a near acceptable speed for any real-world application. This study proposes a proof of concept and provides rationale to further explore the correlation between the mutual information shared between the view and raw data. A function of this process needs to be determined in order to apply a reliable method of mathematical optimisation, potentially allowing the function to logically determine the direction towards the best view.

5 CONCLUSION AND FUTURE WORK

Our preliminary results show that with the variation of a single control measure (in this case, the view orientation), changes in the mutual information between the simulation data and their visualization are quantifiable. This provides a theoretical foundation for our future work in automatic view steering.

The data used in this study were limited to regular structured grid data generated from a simulation in-situ. For direct volume rendering of unstructured and surface meshes, additional voxelization steps can be introduced prior to applying our technique.

Continuing from this study, we will factor in the complete set of parameters controlling the volume render process. These will include view position, focal length and transfer functions.

In particular, we believe transfer functions will have a profound influence on the viewport information due to their integral role in transforming data into visual details such as colour and opacity. Our future work will focus on investigating optimising transfer functions on a per-ray rather than per-dataset basis.

Whilst this study only examined variables bound within a closed domain, the inclusion of additional variables will be infeasible to empirically evaluate the VMI of the complete view. A potential solution to the maximisation problem posed here may be derived from the work of Wells et al. [14]. Further investigation will be conducted to develop a robust method for the maximisation of MI-based model reported here.

ACKNOWLEDGMENTS

The authors wish to thank Robin Richardson and Peter Coveney at University College London for allowing us to use HemeLB for this work.

REFERENCES

- [1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 7 1948. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024>
- [2] M. Chen and H. Jaenicke, "An information-theoretic framework for visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1206–1215, 2010.
- [3] P. Castelló, M. Chover, M. Sbert, and M. Feixas, "Reducing complexity in polygonal meshes with view-based saliency," *Computer Aided Geometric Design*, vol. 31, no. 6, pp. 279–293, 2014.
- [4] J. Gumbau, M. Chover, I. Remolar, and C. Rebollo, "View-dependent pruning for real-time rendering of trees," *Computers & Graphics*, vol. 35, no. 2, pp. 364–374, 2011.
- [5] X. Bonaventura, J. Guo, W. Meng, M. Feixas, X. Zhang, and M. Sbert, "3D shape retrieval using viewpoint information-theoretic measures," *Computer Animation and Virtual Worlds*, vol. 26, no. 2, pp. 147–156, 3 2015. [Online]. Available: <http://doi.wiley.com/10.1002/cav.1566>
- [6] D. Plemenos and M. Benayada, "Intelligent display in scene modelling. New techniques to automatically compute good views." in *GraphiCon '96*, Saint Petersburg (Russia), 7 1996.
- [7] M. Feixas, M. Sbert, and F. González, "A unified information-theoretic framework for viewpoint selection and mesh saliency," *ACM Transactions on Applied Perception (TAP)*, vol. 6, no. 1, p. 1, 2009.
- [8] T. Arbel and F. P. Ferrie, "Viewpoint selection by navigation through entropy maps," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1. IEEE, 1999, pp. 248–254.
- [9] P. Castelló, M. Sbert, M. Chover, and M. Feixas, "Viewpoint-driven simplification using mutual information," *Computers & Graphics*, vol. 32, no. 4, pp. 451–463, 8 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0097849308000551>
- [10] U. D. Bordoloi and H.-W. Shen, "View selection for volume rendering," in *VIS 05. IEEE Visualization, 2005*. IEEE, 2005, pp. 487–494.
- [11] M. Ruiz, I. Boada, M. Feixas, and M. Sbert, "Viewpoint information channel for illustrative volume rendering," *Computers & Graphics*, vol. 34, no. 4, pp. 351–360, 8 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0097849310000439>
- [12] E. G. Learned-Miller, "Entropy and mutual information," *Department of Computer Science, University of Massachusetts, Amherst*, 2013.
- [13] M. Mazzeo and P. Coveney, "HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries," *Computer Physics Communications*, vol. 178, no. 12, pp. 894–914, 6 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465508000805>
- [14] W. M. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Medical Image Analysis*, vol. 1, no. 1, pp. 35–51, 3 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S1361841501800049>